Comp. & Maths. with Appls., Vol. 3, pp. 47–52. Pergamon Press 1977. Printed in Great Britain

NUMERICAL METHODS FOR EIGENSYSTEMS: THE ORR–SOMMERFELD PROBLEM

JOHN M. GERSTRING, JR.
Indiana University-Purdue University at Indianapolis, Indianapolis, Indiana 46205, U.S.A.

Communicated by E. Y. Riding

(*Received September*, 1976)

Abstract—A test problem which may be set up as an initial value, boundary value or eigenvalue problem with adjustable stiffness is presented. It is used to compare five methods (RKI, ORTNRM, SUPORT, GEAR and finite difference) of analysing stiff eigensystems in order to select methods powerful enough to be used effectively on the Orr–Sommerfeld equation. Results are then obtained for plane Poiseuille flow employing the methods found acceptable using the test problem. The plane Poiseuille flow results are then used to further evaluate the methods. Finite difference remains the best algebraic method tested with SUPORT being the best differential method and the least problem dependent.

INTRODUCTION

In earlier work by Gersting and Jankowski[1] the goal was to evaluate solution methods for stiff differential eigensystems, in particular the Orr–Sommerfeld equation. Both differential (direct integration schemes, ORTNRM being the best) and algebraic (finite difference being the best) schemes were examined. The goal here is two-fold, to continue the search for methods and to provide a more suitable vehicle for initial testing of methods than the Orr–Sommerfeld equation which in its usual form is complex and highly stiff.

An example problem consisting of a fourth order linear operator with many of the properties of the Orr–Sommerfeld operator is set up so that it may be used in any of three forms, as an initial value, boundary value or eigenvalue problem. Stiffness in the example is easily controlled and the example has an exact solution, allowing easy comparisons. This example problem can then be used to carry out initial tests on methods in any of the three forms and with any stiffness.

A considerable amount of work has been carried out on methods of analysis of initial value problems[2–4] as well as boundary value problems[5]. Here the discussion will center around the eigenvalue problem. Two recent differential methods are examined and a minor variation in one algebraic method is examined. Comparative times are given for the example problem. The Orr–Sommerfeld equation is then examined using the best method found in the initial comparisons; times and results are given and compared to earlier work.

AN EXAMPLE PROBLEM

Consider the following example problem:

$$\begin{aligned}L(u) = u^{iv} - (3a+b)u''' + (3a(a+b)+c^2)u'' - ((a+b)(a^2+c^2)+2a^2b)u'\\ +ab(a^2+c^2)u = 0\end{aligned}$$

with four consistent conditions selected from

$$\begin{aligned}u(0) &= U(0; c, k)\\ u'(0) &= U'(0; c, k)\\ u''(0) &= U''(0; c, k)\\ u'''(0) &= U'''(0; c, k)\\ u(1) &= U(1; c, k)\\ u'(1) &= U'(1; c, k)\end{aligned} \tag{1}$$

where $U(x; c, k)$ is a function used to prescribe initial and/or boundary conditions: in this case $U(x; c, k)$ will be a specific analytical solution of the system. The independent variable is x; c and k are parameters. The governing equation contains three parameters, a, b, and c. These are specified at the outset except for the case of the eigenvalue problem, where c is considered to be the eigenvalue.

The system $L(u) = 0$ has the characteristic equation

$$(r-a)(r-b)(r-(a+ic))(r-(a-ic)) = 0$$

The roots (or eigenvalues of the Jacobian) are a, b, $a \pm ic$, and the general solution is

$$u_g(x) = c_1 \, e^{ax} + c_2 \, e^{bx} + e^{ax}(c_3 \cos(cx) + c_4 \sin(cx)) \tag{2}$$

To construct a specific solution from the general solution, choose $c_1 = k$, $c_3 = -k$ and $c_2 = c_4 = 0$ resulting in the definition

$$U(x; c, k) = k \, e^{ax}(1 - \cos(cx)) \tag{3}$$

Automatically $u_e(x) = U(x; c, k)$ is the exact analytical solution to system (1).

It is noted that the choice of the constants c_i has eliminated the term e^{bx} from U and theoretically the choice of b has no effect on the solution. Analytically this is true, but computationally (involving finite precision arithmetic) differences can occur if the system becomes stiff.

A common way of describing stiff systems is based on the values of the real parts of the roots of the characteristic equation (also see [6]). If the values differ significantly the system is said to be stiff. Here if all the real parts of a stiff system are negative the system is said to be stiffly stable, if some are positive the system is said to be stiffly unstable. By proper choice of the parameters a and b the stiffness of the system (1) may be controlled. The system may be constituted as one of three types, as an initial value problem (using the four conditions at $x = 0$), as a boundary value problem (using the first two conditions at $x = 0$ and two at $x = 1$), or as an eigenvalue problem (using two homogeneous conditions at $x = 0$ and $x = 1$ with c to be determined).

To demonstrate these two points of interest, variable stiffness and variable problem type, a specific case is examined.

As an initial value problem (1) becomes (with $U(0; c, 1)$)

$$\begin{aligned} L(u) &= 0 \\ u(0) &= 0 \\ u'(0) &= 0 \\ u''(0) &= c^2 \\ u'''(0) &= 3ac^2 \end{aligned}$$

Table 1 shows results for the case $a = -1$, $c = \pi/3$ for various values of b using a standard fixed step size ($h = 0.05$) Runge–Kutta integration scheme. Results are acceptable for the cases $b = -30$ and $b = 1$ but are unsatisfactory for $b = 30$ or 40, the stiffly unstable cases. For larger values of b "overflows" terminate execution.

How to compute accurate results for stiff systems, either initial value, boundary value or eigenvalue problems, is a question on which much work has been done. Evaluations of several methods for initial value problems are given in [2] and [3]. For the linear two-point boundary value problem, e.g. (1) with the first two conditions at $x = 0$ and two conditions at $x = 1$, a usual approach is the method of superposition whereby linearly independent initial value solutions are combined to form the solution to the desired boundary value problem. For stiff systems it may become difficult to maintain the linear independence of the initial value problems used to create the solution. This occurs in the example problem for b larger than about 30. More elaborate integration schemes for the initial value problems forestall but may not eliminate the difficulties.

Table 1. Initial value results for various stiffnesses ($a = -1$, $c = \pi/3$)

x	$u_e = U(x; \pi/3, 1)$ Exact solution	Computational Solutions $b = -30$	$b = 1$	$b = 30$	$b = 40$
0.0	0.0	0.0	0.0	0.0	0.0
0.10	0.00495	0.00495	0.00495	0.00495	0.00495
0.20	0.01789	0.01789	0.01789	0.01789	0.01789
0.30	0.03625	0.03625	0.03625	0.03625	0.03625
0.40	0.05795	0.05795	0.05795	0.05795	0.05795
0.50	0.08126	0.08126	0.08126	0.08126	0.08128
0.60	0.10481	0.10481	0.10481	0.10484	0.10606
0.70	0.12755	0.12755	0.12755	0.12801	0.18844
0.80	0.14867	0.14867	0.14867	0.15752	3.13240
0.90	0.16759	0.16759	0.16759	0.33877	146.370
1.00	0.18394	0.18394	0.18394	3.49560	7164.00

Methods specifically for the analysis of stiff two point boundary value problems have been developed. Two such packages are ORTNRM[7] and SUPORT[8]. ORTNRM is a fixed step process and SUPORT makes use of recent implementations of variable step size initial value integrators. These methods will be examined in the next section.

The still more complicated problem is the eigenvalue problem, e.g. from the example system,

$$
\begin{aligned}
L(u) &= 0 \\
u(0) &= 0 \\
u'(0) &= 0 \\
u(1) &= 0 \\
u'(1) &= 0
\end{aligned}
\tag{4}
$$

The approach taken to the solution of (4) is to fix the values of a and b, estimate a value for the eigenvalue, c, and solve the corresponding boundary value problem using superposition by integrating a set of linearly independent initial value problems across the interval. At $x = 1$ a condition for combining the initial value solutions to satisfy the boundary conditions can be cast as a determinant. An iteration scheme is then employed, in this case Muller's method[9], to refine the estimate for the eigenvalue c by driving the value of the determinant to zero. A typical case was examined using (4) by choosing $a = -1$, an initial estimate of $c = 6$, and choosing a range of values for b to vary the properties from stiffly stable, $b = -30$, through non-stiff, $b = 1$, to stiffly unstable, $b = 50$.

Table 2 gives comparative computing times for four methods of solution to the eigenvalue problem. Each run resulted in a value for c accurate to at least 4 digits, that is, 6.2831. The exact value for c is 2π. The "standard" approach of relating cost (computing time) versus reliability (results for eigenvalue) is used to compare the methods since overall performance is to be evaluated. Points of interest regarding the various methods are examined next.

DISCUSSION

As is to be expected the computing time for the fixed step size Runge–Kutta scheme does not vary with the value of b. Here 20 partitions are used. Results for the RKI method are acceptable until the stiffness parameter b exceeds about 30. At that point the two initial value problems lose their linear independence and the iteration scheme fails. All iterations are carried out using Muller's method. The RKI method has the minimum computing time of all methods but it is noted that doubling of the number of partitions will double the computing time.

The orthonormalized integration technique is an implementation of Conte's work[7] and is described in more detail in [8]. The numbers in parentheses are the number of orthonormalizations required. The "angle" criterion is used to determine when an orthonormalization is required, for these cases it is whenever the angle between the two independent solutions becomes less than 60°. It is noticed that the number of orthonormalizations varies with the stiffness. For

Table 2. CDC 6600 computing times for various methods

b ($a = -1, c$ is EV)	RKI	ORTNRM	GEAR	SUPORT	Finite Difference Muller	EISPAK
−30	0.030	0.263 (34)	0.770	0.146 (0)	0.006 [11]	0.269 [11]
					0.017 [26]	2.594 [26]
					0.063 [101]	
−20	0.030	0.262 (36)	0.700	0.142 (0)		
−10	0.030	0.259 (33)	0.570	0.115 (0)		
−1	0.030	0.245 (25)	0.520	0.095 (0)		
1	0.030	0.249 (26)	0.520	0.100 (0)		
10	0.030	0.284 (54)	0.810	0.170 (0)		
20	0.030	0.310 (77)	1.410	0.275 (1)		
30	0.030	0.315 (87)	1.910	0.418 (2)	0.017 [26]	2.578 [26]
					0.033 [51]	
					0.063 [101]	
40	—	0.320 (90)	2.500	0.512 (2)		
50	—	0.325 (95)	3.050	0.683 (3)		
No. iterations	5	5	5	5	10	—
Special Values	—	Ang = 60°	HO = 10^{-9}	RE = 10^{-4}	—	—
			EPS = 10^{-7}	AE = 10^{-4}	—	—

the case of $b = 50$ orthonormalization occurs at 95 of the possible 100 nodes. Selection of the angle criterion is a critical step in the use of this method; it is heavily dependent on the problem and has a large effect on computing time.

GEAR is an implementation[11] of the variable-order variable-step multistep method developed by Gear[12]. The implementation is for initial value problems (linear or non-linear) and is used here on a set of two initial value problems in the superposition process. The computing times show a definite minimum when the system is not stiff. The Gear method is designed for stiffly stable systems, however it is used here also on the stiffly unstable cases as a test (a user might missapply the method or a system being investigated might change its stiffness properties during an analysis). In the example system when b exceeds about 30 computing times for GEAR begin to rise rapidly. In these tests the Jacobian is not supplied to simulate a case where the user cannot or does not choose to supply it. The implementation then uses a numerical approximation of the partial derivatives required.

SUPORT[8] is a package designed for solution of linear two-point boundary value problems (only slight modifications were required to extract the determinant for the eigenvalue case discussed here). The method of superposition is used along with a Gram–Schmidt orthonormalization process. In overview this package is similar to ORTNRM but with two fundamental differences. The integration schemes provided are variable step size algorithms (RKF Runge–Kutta–Fehlberg is used for the example cases) and the criterion for determining when orthonormalizations are required has been incorporated directly into the reorthonormalization-integration process where it is able to make a good balance in the trade off of orthonormalization vs more integration steps. The times shown for SUPORT are initially lower than all methods except RKI and show a minimum when the system is not stiff. Times increase for the stiffly unstable cases as do the number of orthonormalizations (number in parentheses), both remain reasonable. It is important to note that since a variable step size is employed and the orthonormalization criterion is "built in" this method produces only one set of "times", whereas ORTNRM times could be increased by reducing the step size or changing the angle criterion.

Finally, times for a finite difference scheme are included mainly for comparison and as a connection to earlier work[1, 15]. The times using the Muller iteration process are small. If the eigenvalues are obtained directly from the difference scheme matrices times increase greatly. Both EISPAK[13] and ALLMAT[9] are used, ALLMAT being about 10% slower than EISPAK. Resulting eigenvalues were not as good using the eigenvalue solvers because of the coarse grid (11 or 26 nodes as compared to 101 for the iteration cases, number of nodes is given in square brackets in Table 2).

THE ORR-SOMMERFELD PROBLEM

The reason for constructing the example problem was to create a test vehicle for computational methods of solving eigenvalue problems that would be able to select methods appropriate to analyze highly stiff (in this case stiffly unstable) systems. One such system is the Orr–Sommerfeld equation which arises in fluid mechanics in the study of the hydrodynamic stability of plane Poiseuille flow. The Orr–Sommerfeld problem is

$$\phi^{iv} - 2\alpha^2\phi'' + \alpha^4\phi + i\alpha R((c - \bar{u})(\phi'' - \alpha^2\phi) + \bar{u}''\phi) = 0$$

with boundary conditions

$$\phi(\pm 1) = \phi'(\pm 1) = 0$$

where ϕ is a complex amplitude function, $\bar{u}$ is the primary flow, α is the wave number, R is the Reynolds number and $c = c_r + ic_i$ is the eigenvalue composed of the wave speed, c_r, and the amplification factor, c_i. This system is highly stiffly unstable with characteristic roots including the quantities $\pm\alpha R$.

Table 3 shows comparative times for methods which were successful. CDC 3400 times are from earlier work [1, 14]. As would be expected the RKI method failed. GEAR was tested but was discarded because computing times were excessive (on the order of many minutes on the CDC 6600) as would be indicated by Table 2 for a highly stiffly unstable case. Results for the remaining three methods, ORTNRM, SUPORT and finite difference (using Muller, marked "M", and EISPAK, marked "E") are compared with the work of Thomas [15]. The numbers in parentheses are the number of orthonormalizations required. For all cases the initial estimate for the eigenvalue c is taken as Thomas' result for $R = 1600$, that is, $c = 0.3231 - i\,0.0262$. SUPORT had to be interrupted at $R = 2000$ and restarted to achieve convergence for $R = 2500$; this accounts for the large number of iterations. Finite difference is again included to complete the comparison.

Table 3. Plane Poiseuille flow eigenvalues ($\alpha = 1$, $Re = 2500$)

Method	Time in s/Iteration CDC 3400	Time in s/Iteration CDC6600	Number of partitions	Number of iterations	$c = c_r + ic_i$
ORTNRM	24.5	2.6	200 (90)	10	$0.301148 - i\,0.014179$
SUPORT		10.5	— (2)	15	$0.301150 - i\,0.014199$
F.D. (M)	1.2	0.16	101	10	$0.301149 - i\,0.014182$
F.D. (E)		2.5	26	—	$0.300701 - i\,0.019737$
Thomas	—	—	50	—	$0.3011 \quad - i\,0.0142$

CONCLUSION

The example problem suggested seems to be a good testing ground for solution techniques for eigenvalue problems. The variable stiffness allows each proposed method to be "stretched" to its limit. The example is also useful as a boundary value or initial value problem.

Of the methods examined SUPORT seems to be the strongest. It eliminates many of the "judgement calls" required in ORTRNM (by eliminating the choices of step size and angle criterion) with little added expense. As a boundary value problem solver SUPORT is also extremely powerful. Based on the criteria used in [1], SUPORT would rank before ORTNRM.

No new algebraic methods are discussed here (except for substitution of EISPAK for ALLMAT) so finite difference still heads that list.

Based on computer time alone finite difference would be selected over SUPORT however if implementation time is considered SUPORT is the choice. SUPORT has very few idiosyncrasies and is an extremely good implementation.

Acknowledgements—The author wishes to thank the IUPUI Computer Center for making the computer time available, his students in CS 516 for working with many variations of the example, and Melvin Scott for his help in capturing the determinant in SUPORT.

REFERENCES

1. J. M. Gersting and D. F. Jankowski, Numerical methods for Orr-Sommerfeld problems, *Int. J. Numer. Meth. in Engng* 4, 195–206 (1972).
2. T. E. Hull, W. H. Enright, B. M. Fellen and A. E. Sedgwick, Comparing numerical methods for ordinary differential equations, *SIAM J. Numer. Analysis* 9, 603–637 (1972).

3. W. H. Enright, T. E. Hull and B. Lindberg, Comparing numerical methods for stiff systems of ODE's, *BIT* **15**, 10–48 (1975).
4. L. F. Shampine and M. K. Gordon, *Computer Solution of Ordinary Differential Equations: The Initial Value Problem.* Freeman, San Francisco (1975).
5. M. R. Scott and H. A. Watts, A systematized collection of codes for solving two-point boundary value problems, *Sandia Laboratories Report* SAND75-0539 (1975).
6. R. A. Willoughby (Editor), *Stiff Differential Systems.* Plenum Press, New York (1974).
7. S. D. Conte, The numerical solution of linear boundary value problems, *SIAM Rev.* **8**, 309–321 (1966).
8. M. R. Scott and H. A. Watts, SUPORT—A computer code for two-point boundary-value problems *via* orthonormalization, *Sandia Laboratories Report* SAND75-0198 (1975).
9. R. E. Funderlic (Editor), *The Programmer's Handbook* Union Carbide Corp., Oak Ridge, TN (1968).
10. S. Silverston, ORTNRM—A FORTRAN subroutine package for the solution of linear two-point boundary value problems, *Purdue University, Computer Science Department* Report CSD TR 18 (1968).
11. A. C. Hindmarsh, GEAR: Ordinary differential equation system solver, *Lawrence Livermore Laboratory* Report UCID-30001 (1972).
12. C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations.* Prentice-Hall, Englewood Cliffs (1971).
13. B. S. Garbow, EISPAK—A package for matrix eigensystem routines, *Comp. Phys. Comm.* **7**, 179–184 (1974).
14. J. M. Gersting, Computer programs for the analysis of certain problems in hydrodynamic stability, *Arizona State University Engineering Report* (1970).
15. L. H. Thomas, The stability of plane Pouseuille flow, *Phys. Rev.* **91**, 780–783 (1953).